

AD-A186 646

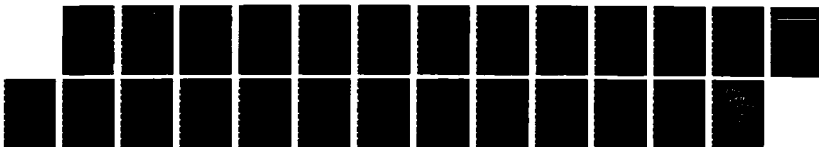
CIRRUS: AN AUTOMATED PROTOCOL ANALYSIS TOOL(U)
CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF PSYCHOLOGY
K VANLEHN ET AL 15 APR 87 PCG-6 N00014-86-K-0149

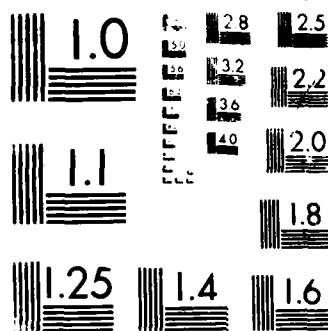
1/1

UNCLASSIFIED

F/G 12/9

NL





RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

DTIC FILE COPY

AD-A186 646

Cirrus: An automated protocol analysis tool

Technical Report PCG-6

Kurt VanLehn and Steve Garlick

Departments of Psychology and Computer Science
Carnegie Mellon Unviersity
Pittsburgh, PA 15213 U.S.A.

DEPARTMENT of PSYCHOLOGY



Carnegie-Mellon University

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

87 10 13 027

12

Cirrus: An automated protocol analysis tool

Technical Report PCG-6

Kurt VanLehn and Steve Garlick

Departments of Psychology and Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 U.S.A.

Appears in P. Langley (ed.)
Proceedings of the Fourth International Workshop on Machine Learning
Los Altos, CA: Kaufman, 1987, pp. 205-217.



This research was supported by Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research, under Contract Numbers N00014-86-K-0349 and N00014-85-C-0688. Approved for public release; distribution limited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) PCG-6			5. MONITORING ORGANIZATION REPORT NUMBER(S) PCG-6	
6a. NAME OF PERFORMING ORGANIZATION Carnegie Mellon University		6b. OFFICE SYMBOL (if applicable)		7a. NAME OF MONITORING ORGANIZATION Personnel and Training Research Program Office of Naval Research (Code 442 PT)
6c. ADDRESS (City, State, and ZIP Code) Psychology Department, BH 335I Schenley Park Pittsburgh, PA 15213		7b. ADDRESS (City, State, and ZIP Code) Arlington, VA 22217		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Same as Monitoring Organization		8b. OFFICE SYMBOL (if applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0349 and N00014-85-C-0688
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO 61153N	PROJECT NO RR04206	TASK NO RR04206-0a WORK UNIT ACCESSION NO NR442a558
11. TITLE (Include Security Classification) Cirrus: an automated protocol analysis tool				
12. PERSONAL AUTHOR(S) VanLehn, Kurt Alan & Garlick, Steve				
13a. TYPE OF REPORT Technical	13b. TIME COVERED FROM 6-15-86 TO 10-31-87	14. DATE OF REPORT Year, Month, Day 87-4-15	15. PAGE COUNT 17	
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	Protocol Analysis, machine learning, decisiontree, student modeling, intelligent tutoring systems	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)				
<p style="text-align: center;">Abstract</p> <p>Cirrus is a tool for protocol analysis. Given an encoded protocol of a subject solving problems, it constructs a model that will produce the same protocol as the subject when it is applied to the same problems. In order to parameterize Cirrus for a task domain, the user must supply it with a problem space: a vocabulary of attributes and values for describing spaces, a set of primitive operators, and a set of macro-operators. Cirrus' model of the subject is a hierarchical plan that is designed to be executed by an agenda-based plan follower. Cirrus' main components are a plan recognizer and a condition inducer. The condition inducer is based on Quinlan's ID3. Preliminary results suggest that Cirrus is robust against noise in the data. Cirrus has potential applications not only in psychology but also as the student modelling component in an intelligent tutoring system.</p>				
20. DISTRIBUTION AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Susan Chipman			22b. TELEPHONE (Include Area Code) (202) 696-4322	22c. OFFICE SYMBOL 4 42 PT

Cirrus: an automated protocol analysis tool

KURT VANLEHN
STEVE GARLICK

VANLEHN@A.PSY.CMU.EDU
GARLICK@A.PSY.CMU.EDU

Department of Psychology, Carnegie-Mellon University,
Pittsburgh, PA 15213 U.S.A.

Abstract

Cirrus is a tool for protocol analysis. Given an encoded protocol of a subject solving problems, it constructs a model that will produce the same protocol as the subject when it is applied to the same problems. In order to parameterize Cirrus for a task domain, the user must supply it with a problem space: a vocabulary of attributes and values for describing spaces, a set of primitive operators, and a set of macro-operators. Cirrus' model of the subject is a hierarchical plan that is designed to be executed by an agenda-based plan follower. Cirrus' main components are a plan recognizer and a condition inducer. The condition inducer is based on Quinlan's ID3. Cirrus has potential applications not only in psychology but also as the student modelling component in an intelligent tutoring system.

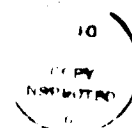
1. Introduction

Cirrus belongs to the class of machine learning programs that induce a problem solving strategy given a set of problem-solution pairs. Other programs in this class are Lex (Mitchell, Utgoff, & Banerji, 1983), ACM (Langley & Ohlsson, 1984), LP (Silver, 1986), and Sierra (VanLehn, 1987). The induction problem solved by these programs is: given a problem space (i.e., a representation for problem states and a set of state-change operators) and a set of problem-solution pairs, find a problem solving strategy such that when the strategy is applied to the given problems, the given solutions are generated.

The primary difference among the programs in this class is the types of problem solving strategies they are designed to induce. Cirrus is unique in that the strategies it produces are hierarchical plans that are designed to be run on an agenda-based plan interpreter. Section 2 describes Cirrus' representation of strategies and discusses the motivations for choosing it.

A secondary difference among strategy induction programs is the amount of information contained in the problem-solution pairs. Some programs (e.g., ACM) expect to be given only the final state while other programs (e.g., LP) expect to be given a sequence of states leading from the initial to the final state. Cirrus expects to be given a sequence of primitive operator applications leading from the initial to the final state. Section 3 describes this aspect of Cirrus' design. This amount of information seems larger than that given to most strategy induction programs. However, because Cirrus assumes a hierarchical plan-following strategy, there may be many changes to the internal control state (i.e., manipulations of the agenda) between adjacent primitive operator applications. These changes are not visible to Cirrus, so it must infer them. The amount of information given to Cirrus in the problem-solution pairs is really not so large after all, given the difficulty of its induction task.

Another major difference among strategy induction programs is their intended application. There seem to be four major applications for strategy induction programs: (1) as models of human skill acquisition, (2) as a way to augment the knowledge base of an expert system, (3) as the student modelling component of an intelligent tutoring system, and (4) as a tool for data analysis. Usually, a program is useful for more than one type of application. Cirrus is intended to be used for the third and fourth applications, i.e., student modelling and data analysis, but it may perhaps prove useful for the others as well. In section 5, we show how Cirrus



A-1

has been employed to construct models of students solving arithmetic problems.

Cirrus' intended applications share the characteristic that the input to Cirrus can be noisy, because it is generated by humans solving problems. People often make slips (Norman, 1981), wherein they perform an action that they didn't intend. A classic example of a slip is an arithmetic error of the type that infest one's checkbook. The existence of slips in the problem-solution pairs means that Cirrus must use induction algorithms, such as Quinlan's (1986) ID3 algorithm, that are fairly immune to noisy data. As far as we know, no other strategy induction program has been used with noisy data.

The main components of Cirrus are a plan recognizer and the ID3 concept inducer. The plan recognizer is a modification of a parser for context-free grammars. Although these components are not particularly novel, their arrangement into a useful system is. Thus, this paper will concentrate on describing the design choices that, we believe, make Cirrus a useful tool. Most of these concern the type of data given Cirrus and the representation of problem solving strategies. These points are discussed in sections 2 and 3. The plan recognizer and concept inducer are discussed in section 4. The performance of Cirrus is illustrated in section 5. Section 6 summarizes and indicates directions for further research.

2. The representation of student models

Cirrus' design assumes that subjects' problem solving strategies are hierarchical. That is, some operations are not performed directly, but are instead achieved by decomposing them into subgoals and achieving those subgoals. For instance, answering a subtraction problem can be achieved by answering each of its columns. A concomitant assumption is that the same goal may be achieved by different methods under different circumstances. For instance, answering a column can be achieved either by taking the difference between the digits in the column when the top digit in the column is larger than the bottom digit, or by first borrowing then taking the difference between the digits when the top digit is smaller than the bottom digit. These two assumptions, that procedural knowledge is hierarchical and conditional, are common to most current theories of human skill acquisition and problem solving (Anderson, 1983; VanLehn, 1983a; Laird, Rosenbloom, & Newell, 1986; Anzai & Simon, 1979).

Another assumption is that goals and the methods for achieving them are schematic and that they must be instantiated by binding their arguments to objects in the problem state. Again, this is a nearly universal assumption among current theories of skill acquisition.

Cirrus employs a simple knowledge representation that is consistent with the above assumptions and introduces very few additional assumptions. The representation is based on two types of operators. *Primitive operators* change the state of the problem. *Macro operators* expand into one or more operators. Table 1 shows a common subtraction procedure in this format.

Both types of operators specify the type of *goal* for which they are appropriate. Goals are just atomic tokens, such as C (for subtract Column) or F (for borrow From) or R (for Regroup). Goals take arguments, which are shown as subscripts in our notation. Thus, the schematic goal C_i represents the general idea of processing a column, and C_2 represents the instantiated goal of processing the tens column (columns are numbered from right to left). The table's notation for macro operators shows the goal to the left of an arrow, and the subgoals to the right of the arrow. The notation indicates argument passing by placing calculations in subscripts of subgoals. Thus, the sixth macro operator means that the goal of processing a regrouping a column (R_i) can be achieved by achieving two subgoals: borrowing from the next column to the left (F_{i-1})

Table 1: Operators for subtraction

Macro Operators	
1. $Sub_i \rightarrow C_i$	Column $i+1$ is not blank
2. $Sub_i \rightarrow C_i$	Column $i+1$ is blank
3. $C_i \rightarrow CT_i$	B_i is blank
4. $C_i \rightarrow \neg_i$	Not $T_i < B_i$
5. $C_i \rightarrow R_i$	$T_i < B_i$
6. $R_i \rightarrow F_{i+1}$	true
7. $F_i \rightarrow S_i$	Not $T_i = 0$
8. $F_i \rightarrow R_i$	$T_i = 0$

Primitive Operators

CT_i	Copy T_i into the answer of column i
\neg_i	Take the difference in column i and write it in the answer
A_i	Add ten to the top of column i
S_i	Put a slash through the top digit of column i
D_i	Decrement the top digit of column i

and adding ten to the current column (A_i).

Macro operators may have associated with them a condition that indicates when the operator is appropriate for achieving its goal. No assumptions are made about the necessity or sufficiency of the given conditions. In table 1, conditions are shown after the subgoals. They are described in English, with the convention that T_i and B_i refer to the top and bottom digits, respectively, of column i . Inside Cirrus, conditions are represented as Lisp predicates.

This representation has been used for many types of procedural knowledge. It is a variant of the notation used in GPS, Strips, and their successors (Nilsson, 1980). To illustrate this generality with a classic example, a macro operator for personal travel might be:

$Travel_{a,b} \rightarrow Travel_{a,Airport(a)} TakePlane_{Airport(a),Airport(b)} Travel_{Airport(b),b}$ if $Distance(a,b) > 500$
 which says that when the distance between points a and b is more than five hundred miles, then a good way to travel between them is to take a plane, which requires travelling from a to the airport near a , and travelling from the airport near b to b .

Although theorists often agree on the hierarchical, conditional nature of procedural knowledge, they usually disagree on how the deployment of this knowledge is controlled. For instance, VanLehn (1983, 1986) assumes that subjects use a last-in-first-out stack of goals, and that they attack subgoals in the order that those subgoals are specified in the macro operators. On the other hand, Newell and Simon (1972), in their discussion of GPS and means-ends analysis, assume that subjects use a goal stack, but they do not always execute the subgoals in a fixed order. Rather, the subjects use a *scheduler* to choose which subgoal of a macro operator to execute first. In principle, the scheduler's choice may be determined by features of the problem state, the goal stack, or whatever memory the solver may have of past actions. The strategy for making such choices is generally assumed to be task-specific, problem-independent knowledge (e.g., that the third subgoal of macro operator five in table 1 should never be scheduled as the initial subgoal).

Yet another control regime is espoused by Anderson and his colleagues (Anderson, 1983; Anderson, Farrell, & Saurers, 1984; Anderson & Thompson, 1986). They assume that instantiated goals are stored in a goal-subgoal tree. Some goals are marked "done" and others are marked "pending." A scheduler may pick

any pending goal from the tree for execution. This tree regime allows the solver to work on the subgoals of one goal for a while, shift to working on another goal's subgoals, then come back to the first goal's subgoals. The stack regimes of Sierra (VanLehn's program) and GPS do not allow such flexibility.

VanLehn and Ball (1987) studied the performances of 26 subjects executing various subtraction procedures and found that a third of them execute their procedures in such a way that the flexibility of the tree regime is necessary for modelling their behavior. For instance, some subjects would first make all the modifications to the top row required by various borrows in the problem, then answer the columns, starting with the leftmost column and finishing with the units column. This execution order requires instantiating all the borrow goals, executing some of their subgoals (the ones that modify the top row), then executing the column-difference subgoals. This requires the ability to temporarily suspend execution of the subgoals of a goal, which the stack regimes do not have.

However, VanLehn and Ball also showed that a simpler control regime, based on maintaining an agenda of goals (i.e., an unordered set of instantiated goals) rather than a tree, is sufficient for modelling their data. Moreover, the agenda regime can do anything that the stack regimes can do, because appropriate scheduling strategies will make an agenda act exactly like a stack. Motivated by this finding, the design of Cirrus assumes that the subject's problem solving strategy is executed by an agenda regime.

The agenda assumption implies that subjects' have two types of procedural knowledge, an operator set (e.g., the one shown in table 1) and a scheduling strategy. The job of a scheduling strategy is to select a goal from the agenda given the current state of the agenda and the current state of the problem. The remainder of this section discusses ways to represent scheduling strategies and motivates the choice of the representation that Cirrus employs.

In principle, the scheduling strategy can be very complicated. Blackboard architectures (Nii, 1986), for instance, approach the scheduling problem with nearly the same complicated, powerful techniques as they employ for attacking the base problem itself. However, one goal of Cirrus is to infer the subjects' scheduling strategies from data, so an unwarranted assumption of complexity and power only makes its job harder without producing better quality models of the subjects. The optimal choice for representation power is a point just slightly beyond that which seems to be minimal, given the data at hand. Thus, when new data are analyzed, Cirrus has a good chance of succeeding and yet the cost of inferring the scheduling strategy is kept reasonable.

To find this optimal point is tricky, because there is a tradeoff in how powerful the scheduling strategy is and how much of the data can be modelled. GPS represented scheduling strategies with a total order on the types of goals (Newell & Simon, 1972, pp. 418 and 436). However, some of the subjects' scheduling choices could not be predicted within this framework.

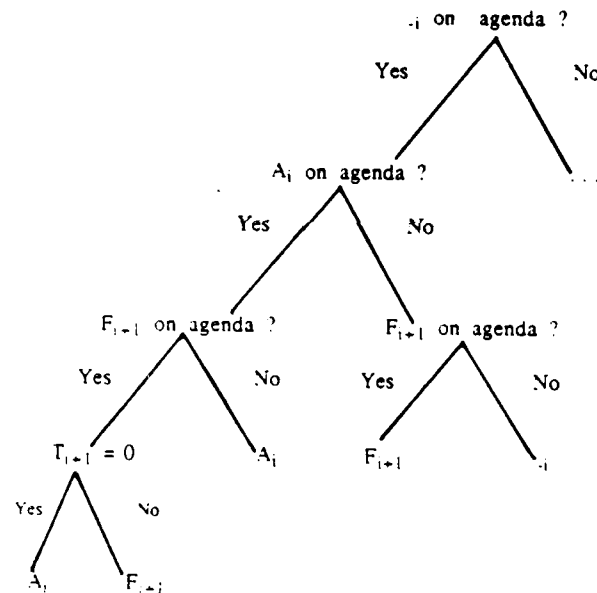
VanLehn and Ball (1987) found that a total order is far too inflexible and that a partial order on goal types allowed much more of the data to be captured. For instance, the two ordering relations, $A_i >_{-1}$ and $R_i >_{-1}$, represent the strategy that the difference in a column must come after the adding of ten to its top digit, but the relative order of the two regrouping subgoals, A_i and F_{i+1} , doesn't matter. VanLehn and Ball found that the best fitting partial orders still did not predict the observed scheduling choices very well. On average, the best fitting partial orders left about a third of the agenda choices underdetermined.

An underdetermined choice means that either the subject truly has no consistent scheduling strategy, or that the subject has a consistent strategy but Cirrus' representation is not adequate to represent it. In order to

avoid the latter case, the scheduling strategy representation should be just powerful enough to represent any subject's strategy. Since it seems unlikely that students' are guessing a third of the time, the Cirrus design uses a representation for scheduling strategies that is slightly more powerful than a partial order on goal types.

Cirrus represents a scheduling strategy with a decision tree, such as the one shown in figure 1. The tree's tests are predicates on the state of the agenda and the state of the problem. The leaves of the tree contain goal types. The scheduler simply traverses the tree, guided by the tests, until it reaches a leaf. Usually, the leaf will contain only one goal type and there will be only one goal of this type on the agenda. If such is the case, the scheduler just selects that goal instance from the agenda; otherwise, the choice is underdetermined. The decision tree shown in figure 1 represents the scheduling heuristics $A_i >_{-i}$ and $F_{i+1} >_{-i}$, but in addition, it represents the heuristic that one should do F_{i+1} first unless the top digit in column $i+1$ is zero. The latter heuristic can not be represented by a partial order on goal types.

Figure 1: A decision tree for a scheduling strategy



In summary, the assumptions built into the design of Cirrus entail that the representation of knowledge has two components: a set of operators, and a decision tree. The operators represented knowledge about how to decompose goals into subgoals and when such decompositions are appropriate. The decision tree represents the subject's strategy for when to work on what types of goals.

3. Cirrus' inputs and outputs

Cirrus takes as inputs (1) a set of problem-solution pairs generated by someone solving problems, (2) a vocabulary for describing problem states, and (3) a set of operators. The latter two inputs are sometimes referred to as a problem space. Cirrus produces a decision tree that represents the subject's scheduling strategy. This section describes these inputs and outputs, and discusses the design issues behind them.

In a problem-solution pair, the solution is represented by an *encoded protocol*, which is a sequence of primitive operator applications. For instance, the first few steps of solving a subtraction problem might be encoded as $[A_1, -_1, S_2, D_2 \dots]$. This sequence represents the actions of adding ten to the top digit in the units column, then taking the units column difference and entering it in the answer, then putting a slash through the top digit of the tens column, then decrementing the top digit of the tens column by one and writing the result above the tens column.

Such encoded protocols may be hand-generated by a human reading a transcript of an experiment, or they may be produced automatically by, for instance, recording the user's selections from a menu-driven interface to an experimental apparatus or an intelligent tutoring system.¹ Other strategy induction programs, such as ACM (Langley & Ohlsson, 1984), assume that only the final state (e.g., the answer to a subtraction problem) is available. The advantage of that assumption is that it is much easier to hand-encode a final state than to hand-encode a whole protocol. However, we believe that the state of the art in personal computing and user interfaces is such that automatic generation of encoded protocols will become so common that hand-encoding will soon become a lost art. Thus, there will be no penalty for assuming that encoded protocols are the input.

The second input to Cirrus is a vocabulary of attributes for describing problem states. These attributes are used as tests in the decision tree that Cirrus builds. An attribute can be any discrete-valued function on a state. If all the functions are predicates, then Cirrus will build only binary trees; however, it can handle functions with any type of range.² An attribute is a Lisp function with three inputs: the external problem state (e.g., a partially solved subtraction problem), the agenda, and the list of the operator applications up until this point. The latter argument is needed in order to implement attributes such as "Has there been a borrow in this problem yet?" or "Was the last primitive operator a decrement?"

As usual when dealing with induction programs, the user must experiment with attribute vocabularies until the program begins to generate sensible results. This is why we consider Cirrus a *tool* for data analysis, and not an automated psychologist. The art of analysing protocols lies in intuiting what the subject thinks is relevant. Such intuitions are encoded as attribute vocabularies, and Cirrus is used to do the bookkeeping involved in checking them against the data. The ultimate goal of protocol analysis is a model that is consistent with the data, parsimonious and psychologically plausible. Cirrus can evaluate the consistency of the model, but only a human can evaluate psychological plausibility. Parsimony could in principle be measured by Cirrus, but that is left to the human at present.

When the combination of human and Cirrus have found an attribute vocabulary that seems to work for a sample of the subject population, then it may be reasonable to assume that it will continue to yield good

¹The tests of Cirrus reported here use hand-encoded protocols from the VanLehn and Ball (1987) experiments.

²Only values that actually occur during the tree building are placed in the tree, so attribute functions will work even if they have infinite ranges. Of course, one would probably want to include a mechanism that breaks large ranges into intervals, in order to enhance the generality of the induced scheduling strategy.

models for subjects who have not yet been tested. If so, then Cirrus can now be used as an automated model builder. Such model builders are used in intelligent tutoring systems in order to track the learning of a student so that the system can offer appropriate instruction. Cirrus can be used as the student modelling module in such a system, but it must first be parameterized for a task domain by empirically deriving the appropriate attribute vocabulary.

The output of Cirrus is a subject model, which consists of an operator set and a decision tree, as previously described. The subject model will be consistent with the set of protocols that it is given. Here, "consistency" means that the model will generate exactly the same primitive operator applications as the subjects, if the model is deterministic. If the model is underdetermined, say, because some of the leaves in the decision tree have more than one goal type in them, then one of the possible solution paths generated by the model for each problem must correspond to the subject's protocol.

Currently, Cirrus does not build the whole subject model, but only the decision tree. The operator set is given to it. This has not been a problem in the initial application because we have a theory of how arithmetic is learned (VanLehn, 1983a; VanLehn, 1983b) that predicts the operator sets that subjects will have. For subtraction, for instance, the theory predicts that the subjects will have one of 30 operator sets (VanLehn, 1983b). In principle, Cirrus could iterate through these 30 possibilities and choose the one that maximize the fit and parsimony of the subject models. However, this choice is currently done by the user.

In task domains where no theory is available, Cirrus could be augmented to accept even less information about the operator sets. It could take only the goals and subgoals of operators as input, and use standard machine learning techniques (Langley et al., 1980) to induce the conditions of the operators.

If one is studying skill acquisition, then making a plausible assumptions about the goals and subgoals of operators is not as hard a problem as it may seem. It has often been observed that instruction tells the student what to do (i.e., what subgoals go with each goals) but not when to do it (Anderson, 1983; VanLehn, 1983b). In cases where the instruction does mention the bodies of operators, then one can generate a plausible set of operator sets by introducing perturbations (e.g., deleting subgoals from operators) into the operator set that the material teaches.

4. Computational Details of Cirrus

Cirrus performs the following operations in order:

1. *Plan recognition.* Cirrus parses each protocol using the given operator set as if it were a context-free grammar. This converts the protocol, which is a sequence of primitive operator applications, into a *parse tree* whose leaves are the primitive operator applications and whose interior nodes are macro operator applications.
2. *Tree walking.* The parse tree is traversed depth-first in order to infer what the state of the agenda must have been at each cycle of execution. The output of tree walking is a *micro state protocol*, which is the original protocol augmented by all the agenda states and state changes that happen between the changes in the external problem state. This protocol represents what one would see if one could see the subject's internal actions as well as the external ones.
3. *Attribute encoding.* In preparation for building a decision tree, Cirrus collects all instance of agenda selections. These are represented initially as the selected item paired with the total state (i.e., the problem, the agenda, and the preceding operator applications) at the time of the selection. Each total state is converted to an *attribute vector* by running all the attribute functions on it. The output of this stage is a set of pairs, each consisting of an agenda selection and the attribute vector for the total state at the time of the agenda selection.

4. *Decision Tree construction.* The last stage is to build a decision tree that will sort the set of pairs by the goal type of the agenda selection. Cirrus uses Quinlan's (1986) ID3 algorithm. Although we included Quinlan's Chi-squared test (op. cit, pg. 93) hoping that it would prevent ID3 from trying to predict the occurrence of noise (i.e., slips), we realize now that it is inappropriate for this purpose. A slip tends to show up as a single instance of an agenda selection that is not of the same type as all the other agenda selections in the set. The Chi-squared test is not an appropriate test for detecting an exception. It is good for taking a heterogeneous set and showing that any way one discriminates it, the resulting partitions are just as heterogeneous as the original set, so partitioning the set is pointless. However, with a set that is homogeneous with one exception, virtually any discrimination helps, so slips almost always remain undetected by the Chi-squared test.

The use of a parser for plan recognition deserves some discussion, for it is not as common an AI technique as decision tree building, which is the only other nontrivial operation performed by Cirrus. If an ordinary context-free parser built the parse trees, then a parse tree would obey the following constraints (for easy reference, we say that A is a "suboperator" of B if A appears just beneath B in the parse tree):

1. If two operators are suboperators of the same macro operator, then their order in the parse tree is the same as their order in the right side of the macro operator's rule.
2. If two operators are suboperators of the same macro operator, and they are adjacent in the parse tree, then the portion of the protocol that each covers must *not overlap*. (This is guaranteed by the fact that a parse tree is a tree, in that operators have just one parent in the tree.)
3. If two operators are suboperators of the same macro operator, and they are adjacent in the parse tree, then the portion of the protocol that each covers must be *contiguous*. There can not be primitive operator applications between them.

If the control regime were the deterministic stack regime used by Sierra, then this kind of parsing would be appropriate. If the control regime were the scheduled stack regime used by GPS, then relaxing the first constraint would be necessary. Because Cirrus uses an agenda control regime, both the first and third constraints must be relaxed. The third constraint must be removed because the agenda regime makes it possible to stop working on one macro operator's subgoals, work on some other macro operator's subgoals, then go back to the original macro operator's subgoals. This implies that suboperators of the same macro operator need not abut.

Relaxing these constraints is a minor change to the parsing algorithm, but it drastically increases the search space for parse trees. Consequently, we added more assumptions to Cirrus. It is assumed that (1) the values of arguments can not be changed once the goal has been instantiated, and (2) the values of a subgoal's arguments are calculated exclusively from the values of the goal's arguments, regardless of the rest of the total state at the time of instantiation. This makes the parser appropriate for attribute grammars (Knuth, 1968) and affix grammars (Koster, 1971) in that it can use the argument values of the goals in order to constrain the search. This vastly improves the performance of the parser, at least for subtraction.

One constraint that the parser does not use, although it could, is provided by the conditions on the macro operators. These could be tested during parsing in order to prune parse trees where macro operators were applied inappropriately. However, since we intend that Cirrus eventually be able to induce these conditions or modify existing ones, we designed the parser to ignore them.

Lastly, the parser is very similar to the one used in Sierra (VanLehn, 1987). Sierra can learn new macro-operators from protocols. Thus, it is possible to modify Cirrus so that when it is given an operator set that is incomplete, it can discover new operators. It is not clear whether this is a useful feature in the applications for which Cirrus is intended.

Figure 2: Paul's problem solutions (above) and protocols (below)

1 $\begin{array}{r} 647 \\ - 45 \\ \hline 602 \\ 02 \end{array}$	2 $\begin{array}{r} 8305 \\ - 3 \\ \hline 8302 \end{array}$	3 $\begin{array}{r} 885 \\ - 205 \\ \hline 680 \end{array}$	4 $\begin{array}{r} 713 \\ - 44 \\ \hline 39 \end{array}$
5 $\begin{array}{r} 410 \\ - 23 \\ \hline 27 \end{array}$	6 $\begin{array}{r} 512 \\ - 3 \\ \hline 559 \end{array}$	7 $\begin{array}{r} 1418 \\ 5 \cancel{4} \cancel{8} 11 \\ - 2697 \\ \hline 3894 \end{array}$	8 $\begin{array}{r} 10 \\ 2 \cancel{0} 11 \\ - 214 \\ \hline 097 \end{array}$
9 $\begin{array}{r} 10 \\ 7 \cancel{0} 13 \\ - 1 \cancel{8} \cancel{1} 3 \\ \hline 215 \\ 1598 \end{array}$	10 $\begin{array}{r} 310015 \\ - 607 \\ \hline 3408 \end{array}$	11 $\begin{array}{r} 9910 \\ 0 \cancel{10} \cancel{10} \cancel{0} 12 \\ - \cancel{1} \cancel{0} \cancel{0} \cancel{1} \cancel{2} \\ \hline 214 \\ 9798 \end{array}$	12 $\begin{array}{r} 99 \\ 7 \cancel{10} \cancel{10} 11 \\ - \cancel{8} \cancel{0} \cancel{0} \cancel{1} \\ \hline 43 \\ 7958 \end{array}$

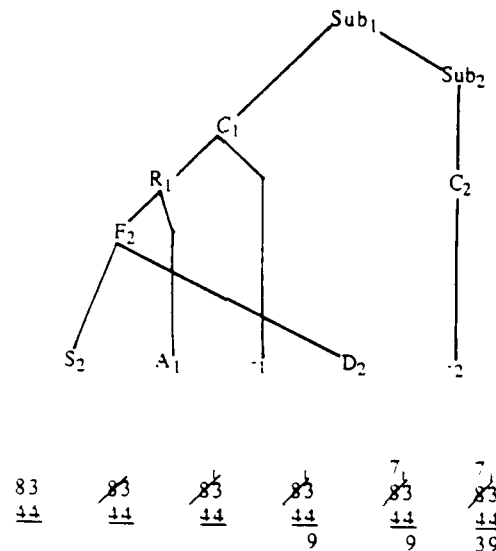
1. $\neg_1 \neg_2 CT_3$
2. $\neg_1 CT_2 CT_3 CT_4$
3. $\neg_1 \neg_2 \neg_3$
4. $S_2 A_1 \neg_1 D_2 \neg_2$
5. $S_2 A_1 \neg_1 D_2 \neg_2$
6. $S_2 A_1 \neg_1 D_2 CT_2 CT_3$
7. $S_2 A_1 \neg_1 D_2 S_3 A_2 D_3 \neg_2 S_4 A_3 \neg_3 D_4 \neg_4$
8. $S_2 A_1 \neg_1 D_2 S_3 A_2 \neg_2 D_3 \neg_3$
9. $S_2 A_1 \neg_1 D_2 S_3 A_2 \neg_2 D_3 \neg_3 CT_4$
10. $S_2 A_1 \neg_1 D_2 \neg_2 S_4 A_3 D_4 \neg_3 \neg_4$
11. $S_2 A_1 \neg_1 D_2 S_5 D_5 A_4 S_4 D_4 A_3 S_3 D_3 A_2 \neg_2 \neg_3 CT_4 CT_5$
12. $S_4 D_4 A_3 S_3 D_3 A_2 S_2 D_2 A_1 \neg_1 \neg_2 \neg_3 \neg_4$

5. Analyzing a subtraction protocol

This section illustrates the operation of Cirrus by describing the analysis of a protocol from Paul, a third grade student who has just learned subtraction. Paul worked a twelve-item test while we recorded his writing actions. Figure 2 shows his solution to the problems, first as a worked problem and second as a sequence of primitive operator applications. During the encoding process, we filtered out one recognizable slip. Paul

wrote his answer to problem 1 illegibly, so he went back and rewrote it. These actions are not included in the encoded protocol. Although this particular instance of "editing" the data is quite clearly appropriate, part of the process of understanding a protocol may involve conjecturing that certain actions are slips and changing them to what the user thinks the subject intended to do. If this makes Cirrus generate a dramatically better analysis, the conjecture is supported.

Figure 3: The parse tree for Paul's problem four

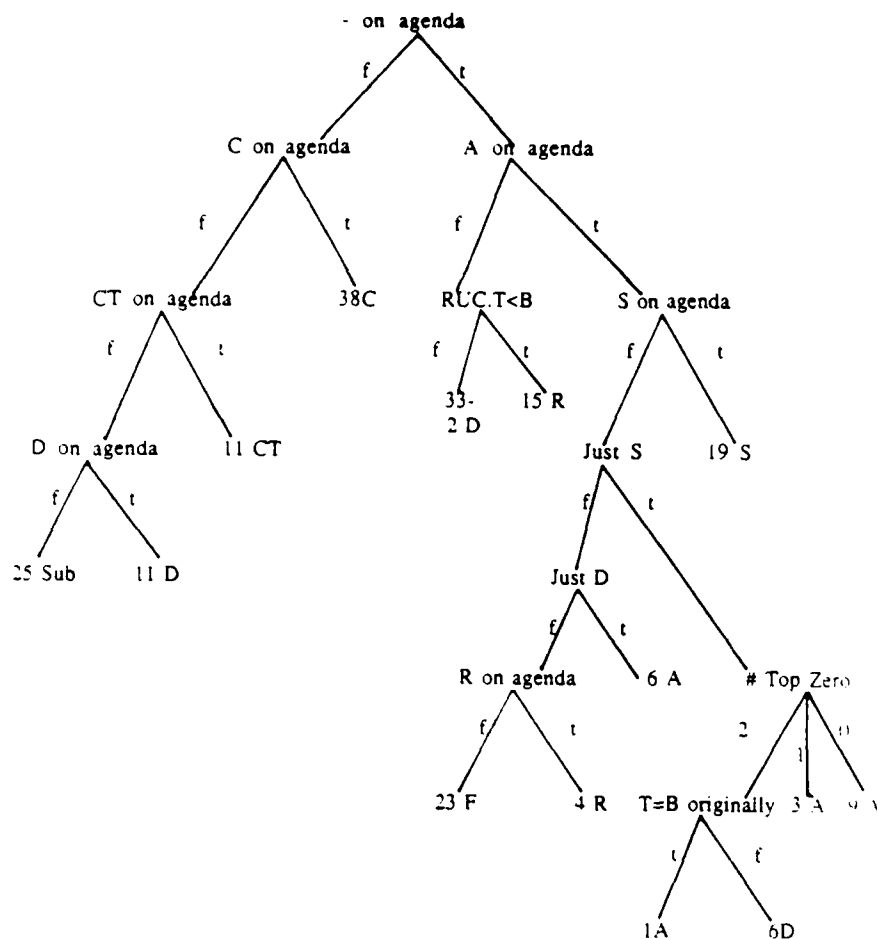


The protocols are parsed, using the grammar of table 1. This yields a set of parse trees. Figure 3 shows the parse tree for problem 4. The parse trees are walked, the attribute vectors are collected, and a decision tree is produced. Figure 4 shows the decision tree that represents Paul's scheduling strategy. The leaves of the tree show how many agenda selections of each type have been sorted to that leaf.

Paul's most common strategy for borrowing is to do the scratch mark, then the add ten and column difference, then return to complete the borrowing by doing a decrement (see figure 3). However, on some columns, Paul does the standard strategy of scheduling the decrement so that it immediately follows the scratch mark. These two strategies have much in common, and most of the decision tree is devoted to representing these commonalities. The most interesting part of the tree is the subtree just to the right of the node labelled "Just S". This subtree encodes the sub-strategy for what to do just after a scratch mark has been performed. It represents Cirrus' best guess about the conditions that trigger Paul's two strategies. The attributes chosen have to do with the existence of zeros in the top row of the problem. Apparently, for borrow-from-zero problems (problems 11 and 12), Paul prefers to use the strategy he was taught rather than the strategy he invented. On the basis of this intuition, the user would probably want to augment the attribute vocabulary with an attribute that indicates a column requires borrowing from zeros, because the attributes currently available to Cirrus are such that it must use an implausible attribute, $T=B$ originally, in order to

Figure 4: Paul's decision tree.

"Just X" means that action X was the most recently executed primitive operator.
 "RUC.T<B" means that the top digit is less than the bottom in the rightmost unanswered column.
 "T=B originally" means that the column's digits were equal in the initial problem state.
 "# top zeros" counts the number of zeros in the top row.



differentiate the tens column of problem 11 from the others.³ This symbiotic exploration of hypotheses about cognition is exactly what Cirrus is designed to expedite.

Although experimentation with attribute sets is inevitable with any induction algorithm, this particular example indicates a problem that seems to be unique to ID3. The desired attribute, which should distinguish borrow-from-zero columns from other kinds of columns, is a conjunction of two existing attributes, $T_i < B_i$ and $T_{i+1} = 0$. We had thought that ID3 would capture this conjunction as a subtree with two tests, say, $T_i < B_i$ on the upper node and $T_{i+1} = 0$ on the lower node. However, the $T_i < B_i$ test is always true at this point in the tree, so it has no discriminatory power; hence, ID3 would never choose it. This seems like a major problem

³A similar situation exists at the leaf under the false branch of RUC.T<B. Although the leaf is shown as containing two types of agenda selections, 33 occurrences of take-difference and 2 occurrences of decrement, Cirrus actually placed a large subtree here filled with implausible attributes. Again, straightening out this subtree would require experimentation with the attribute vocabulary.

with ID3. Its only cure may be to augment the original attribute vocabulary with all possible conjunctions of attributes, a combinatorially infeasible solution. This suggests that ID3 may not be the right choice for an induction algorithm, a possibility we wish to explore in further research.

6. Directions for further research

The only way to evaluate the utility of Cirrus as a tool is to use it as such. This is the first item on our agenda for future research. Our hope is that the combination of Cirrus and human analyst is a more productive combination than human analyst alone. In particular, we will be reanalyzing the rest of the VanLehn and Ball protocols in the hope that Cirrus will help uncover patterns that were overlooked by the unaided human theorists.

As mentioned earlier, we intend to have Cirrus induce the conditions on macro-operators instead of having the user supply them.

There is nothing in our theory of skill acquisition that stipulates that ID3 is the right model of scheduling or macro-operator conditions. (Indeed, we find the decision tree format a difficult one to understand -- a common complaint about decision trees (Quinlan, 1986).) We would like to try a variety of concept formation techniques. We are particularly interested in exploring techniques based on models of human concept acquisition (Smith & Medin, 1981). If people learned their conditions and strategies by induction, then using the same inductive mechanisms, with the same built-in biases, seems like a good way to infer those conditions.

Lastly, there is a sense in which Cirrus is a theory of problem solving. If Cirrus fails to analyze a protocol, then the protocol lies outside its theory of problem solving.⁴ There are two ways analysis can fail: (1) The parser may be unable to parse some of the protocol. This indicates either that the operator set is wrong (i.e., a parameter to the theory has the wrong value -- a minor problem) or that the subject is using a more powerful plan following regime. It would be interesting to use Cirrus as a filter on large quantities of protocol data, looking for subjects who are using a more powerful control regime. (2) The second kind of failure occurs when the decision tree builder builds unintelligible trees. This could be the fault of the attribute vocabulary (again, a minor problem with an incorrectly set parameter), or more interestingly, the theory of scheduling strategies could be wrong. We actually believe that the latter may be the case, and look forward to finding evidence for it by uncovering protocols that Cirrus can not analyze.

Acknowledgments

This research was supported by the Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research, under Contract Numbers N00014-86-K-0349 and N00014-85-C-0688.

⁴Cirrus' theory of problem solving is discussed in (Garlick & VanLehn, 1987). See Bashkar and Simon's (1977) for an example of a protocol analyzer with a strong theory built into it.

References

- Anderson, J. R. *The Architecture of Cognition*. Cambridge, MA: Harvard, 1983.
- Anderson, J.R. & Thompson, R. Use of analogy in a production system architecture. 1986. Paper presented at the Illinois Workshop on Similarity and Analogy, Champaign-Urbana, June, 1986.
- Anderson, J. R., Farrell, R., & Saurers, R. Learning to program in LISP. *Cognitive Science*, 1984, 8, 87-129.
- Anzai, Y. & Simon, H.A. The theory of learning by doing. *Psychological Review*, 1979, 86, 124-140.
- Bhaskar, R. & Simon, H. A. Problem solving in a semantically rich domain: An example from engineering thermodynamics. *Cognitive Science*, 1977, 1, 193-215.
- Garlick, S. & VanLehn, K. *Deriving descriptions of the mind: A rationale for serial models of cognition*. (Technical Report PCG-6). Carnegie-Mellon University, Dept. of Psychology, 1987.
- Knuth, D.E. Semantics of context-free languages. *Mathematical Systems Theory*, 1968, 2, 127-145.
- Koster, C.H.A. Affix grammars. In J.E. Peck (Ed.), *ALGOL 68 Implementation*. Amsterdam: North-Holland, 1971.
- Laird, J., Rosenbloom, P. & Newell, A. The generality of a simple learning mechanism: Chunking in SOAR. *Journal of Machine Learning*, 1986.
- Langley, P. & Ohlsson, S. Automated cognitive modeling. In *Proceedings of AAAI-84*. Los Altos, CA: Morgan Kaufman, 1984.
- Langley, P., Neches, R., Neves, D. & Anzai, Y. A domain-independent framework for procedure learning. *Policy Analysis and Information Systems*, 1980, 4, 163-197.
- Mitchell, T. M., Utgoff, P. E. & Banerji, R. B. Learning problem-solving heuristics by experimentation. In R. S. Michalski, T. M. Mitchell & J. Carbonell (Eds.), *Machine Learning*. Palo Alto, CA: Tioga Press, 1983.
- Newell, A. & Simon, H. A. *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall, 1972.
- Nii, P. The blackboard model of problem solving. *AI Magazine*, Summer 1986, 7(2), 38-53.
- Nilsson, N. *Principles of Artificial Intelligence*. Los Altos, CA: Morgan Kaufman, 1980.
- Norman, D. A. Categorization of action slips. *Psychological Review*, 1981, 88, 1-15.
- Quinlan, J. R. The effect of noise on concept learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach. Volume II*. Los Altos, CA: Morgan Kaufman, 1986.
- Silver, B. Precondition analysis: Learning control information. In Michalski, R.S., Carbonell, J.G. & Mitchell, T.M. (Eds.), *Machine Learning: An Artificial Intelligence Approach. Vol. 2*. Los Altos, CA: Morgan-Kaufman, 1986.
- Smith, E.E. & Medin, D. *Categories and concepts*. Cambridge, MA: Harvard University Press, 1981.
- VanLehn, K. Human skill acquisition: Theory, model and psychological validation. In *Proceedings of AAAI-83*. Los Altos, CA: Morgan Kaufman, 1983.
- VanLehn, K. *Felicity conditions for human skill acquisition: Validating an AI-based theory* (Tech. Report CIS-21). Xerox Palo Alto Research Center, 1983.
- VanLehn, K. Learning one subprocedure per lesson. *Artificial Intelligence*, 1987, 31(1), 1-40.
- VanLehn, K. & Ball, W. *Flexible execution of cognitive procedures* (Technical Report PCG-5). Carnegie-Mellon University, Dept. of Psychology, 1987.

Dr. Phillip L. Ackerman
University of Minnesota
Department of Psychology
Minneapolis, MN 55455

Dr. Beth Adelson
Department of Computer Science
Tufts University
Medford, MA 02155

Dr. Robert Ahlers
Code N711
Human Factors Laboratory
Naval Training Systems Center
Orlando, FL 32813

Dr. Ed Aiken
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Robert Aiken
Temple University
School of Business Administration
Department of Computer and
Information Sciences
Philadelphia, PA 19122

Dr. James Algine
University of Florida
Gainesville, FL 32605

Dr. John Allen
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. William E. Alley
AFMRL/MOT
Brooks AFB, TX 78235

Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Thomas H. Anderson
Center for the Study of Reading
174 Children's Research Center
51 Gerty Drive
Champaign, IL 61820

Dr. Steve Andriole
George Mason University
School of Information
Technology & Engineering
4400 University Drive
Fairfax, VA 22030

Technical Director, ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Alan Baddley
Medical Research Council
Applied Psychology Unit
15 Chaucer Road
Cambridge CB2 2EF
ENGLAND

Dr. Patricia Baggett
University of Colorado
Department of Psychology
Box 345
Boulder, CO 80309

Dr. Eva L. Baker
UCLA Center for the Study
of Evaluation
145 Moore Hall
University of California
Los Angeles, CA 90024

Dr. Meryl S. Baker
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Isaac Bejar
Educational Testing Service
Princeton, NJ 08540

Leo Beltracchi
United States Nuclear
Regulatory Commission
Washington, DC 20555

Dr. Mark H. Bickhard
University of Texas
EDB 504 ED Psych
Austin, TX 78712

Dr. John Black
Teachers College
Columbia University
525 West 121st Street
New York, NY 10027

Dr. Arthur S. Blawies
Code N711
Naval Training Systems Center
Orlando, FL 32813

Dr. Robert Blanchard
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. R. Darrell Bock
University of Chicago
NORC
6030 South Ellis
Chicago, IL 60637

Dr. Jeff Bonar
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Richard Braby
NTSC Code 10
Orlando, FL 32751

Dr. Jomilla H. Braddock II
Center for the Social
Organization of Schools
The Johns Hopkins University
3505 North Charles Street
Baltimore, MD 21218

Dr. Robert Breux
Code N-095R
Naval Training Systems Center
Orlando, FL 32813

Dr. Ann Brown
Center for the Study of Reading
University of Illinois
51 Gerty Drive
Champaign, IL 61820

Commanding Officer
CAPT Lorin W. Brown
NROTC Unit
Illinois Institute of Technology
3300 S. Federal Street
Chicago, IL 60618-3793

Dr. John S. Brown
XEROX Palo Alto Research
Center
3333 Coyote Road
Palo Alto, CA 94304

Dr. John Bruer
The James S. McDonnell
Foundation
University Club Tower, Suite 1610
1034 South Brentwood Blvd
St. Louis, MO 63117

Dr. Bruce Buchanan
Computer Science Department
Stanford University
Stanford, CA 94305

Dr. Patricia A. Butler
OERI
555 New Jersey Ave. NW
Washington, DC 20208

Dr. Tom Cafferty
Dept. of Psychology
University of South Carolina
Columbia, SC 29208

Dr. Joseph C. Campione
Center for the Study of Reading
University of Illinois
51 Gerty Drive
Champaign, IL 61820

Joanne Capper
Center for Research into Practice
1718 Connecticut Ave. NW
Washington, DC 20009

Dr. Susan Carey
Harvard Graduate School of
Education
337 Gutman Library
Appian Way
Cambridge, MA 02138

Dr. Pat Carpenter Carnegie-Mellon University Department of Psychology Pittsburgh, PA 15213	Dr. William Clancey Stanford University Knowledge Systems Laboratory 701 Welch Road, Bldg C Palo Alto, CA 94304	Dr. Thomas M. Duffy Communications Design Center Carnegie-Mellon University Schlenley Park Pittsburgh, PA 15213
Dr. John M. Carroll IBM Watson Research Center User Interface Institute P O Box 218 Yorktown Heights, NY 10590	Dr. Charles Clifton Tobin Hall Department of Psychology University of Massachusetts Amherst, MA 01003	Dr. Richard Duran University of California Santa Barbara, CA 93106
LCDR Robert Carter Office of the Chief of Naval Operations OP-01B Pentagon Washington, DC 20350 2000	Dr. Allan M. Collins Boft Berenick & Newman Inc 50 Moulton Street Cambridge MA 02130	Edward E. Eddowes CNATRA N301 Naval Air Station Corpus Christi, TX 78419
Dr. Alphonse Chapenis 8415 Bellona Lane Suite 210 Buxton Towers Baltimore, MD 21204	Dr. Stanley Collier Office of Naval Technology Code 222 800 N. Quincy Street Arlington, VA 22217-5000	Dr. John Ellis Navy Personnel RAD Center San Diego, CA 92252
Dr. Davide Charney English Department Penn State University University Park, PA 16802	Dr. Lynn A. Cooper Learning RAD Center University of Pittsburgh 3939 O'Hare Street Pittsburgh, PA 15213	Dr. Jeffrey Elman University of California San Diego Department of Linguistics C-008 La Jolla, CA 92093
Dr. Paul R. Chatelier OUSDRE Pentagon Washington, DC 20350-2000	LT Judy Crookshanks Chief of Naval Operations OP-112G5 Washington, DC 20350-2000	Dr. Susan Embretson University of Kansas Psychology Department 426 Fraser Lawrence, KS 66045
Dr. Michelene Chi Learning R & D Center University of Pittsburgh 3939 O'Hare Street Pittsburgh, PA 15213	Phil Cunniff Commanding Officer Code 7522 Naval Undersea Warfare Engineering Keyport, WA 98345	Dr. Randy Engle Department of Psychology University of South Carolina Columbia, SC 29208
Dr. L. J. Chmura Computer Science and Systems Code 7590 Information Technology Division Naval Research Laboratory Washington, DC 20375	Dr. Cary Czichon Intelligent Instructional Texas Instruments AI Lab P O Box 660245 Dallas, TX 75266	Dr. William Epstein University of Wisconsin W J Bragden Psychology Bldg 1202 W Johnson Street Madison, WI 53706
Mr. Raymond E. Christel AFHRL/MOE Brooks AFB, TX 78235	Brian Dallman 3400 TTW-TTCXS Lowry AFB, CO 80230-5000	ERIC Facility-Acquisitions 4833 Rugby Avenue Bethesda, MD 20014
	Dr. Gerald F. DeJong Artificial Intelligence Group Coordinated Science Laboratory University of Illinois Urbana, IL 61801	Dr. K. Anders Ericsson University of Colorado Department of Psychology Boulder, CO 80309
	Goery Delacote Directeur de L'informatique Scientifique et Technique CNRS 15 Quai Anatole France 75700 Paris FRANCE	
	Dr. Thomas E. DeZern Project Engineer, AI General Dynamics PO Box 748 Fort Worth, TX 76101	
	Dr. Andrea di Sessa University of California School of Education Tolman Hall Berkeley, CA 94720	
	Dr. R. K. Dismukes Associate Director for Life Sciences AFOSR Bolling AFB Washington, DC 20332	
	Dr. Stephanie Doan Code 6021 Naval Air Development Center Warminster, PA 18973-5000	
	Defense Technical Information Center Cameron Station Bldg 5 Alexandria, VA 22304 Attn TC (12 Copies)	
	Dr. Cary Crichon Intelligent Instructional Systems Texas Instruments AI Lab P O Box 660245 Dallas, TX 75266	

Dr Beatrice J Farr
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr Marshall J Farr
Farr-Sight Co
2520 North Vernon Street
Arlington, VA 22207

Dr Paul Fellovich
Southern Illinois University
School of Medicine
Medical Education Department
P O Box 3926
Springfield IL 62708

Mr Wallace Feurzeig
Educational Technology
Bolt Beranek & Newman
10 Moulton St
Cambridge MA 02238

Dr Gerhard Fischer
University of Colorado
Department of Computer Science
Boulder CO 80309

J D Fletcher
9931 Corsica Street
Vienna VA 22180

Dr Linda Flower
Carnegie-Mellon University
Department of English
Pittsburgh, PA 15213

Dr Kenneth D Forbus
University of Illinois
Department of Computer Science
1304 West Springfield Avenue
Urbana IL 61801

Dr Barbara A Fox
University of Colorado
Department of Linguistics
Boulder CO 80309

Dr Carl H Frederiksen
McGill University
3700 McTavish Street
Montreal Quebec H3A 1Y2
CANADA

Dr John R Frederiksen
Bolt Beranek & Newman
50 Moulton Street
Cambridge MA 02138

Dr Michael Genesereth
Stanford University
Computer Science Department
Stanford, CA 94305

Dr Dedre Gentner
University of Illinois
Department of Psychology
603 E Daniel St
Champaign IL 61820

Lee Gladwin
Route 3 -- Box 225
Winchester VA 22601

Dr Robert Glaser
Learning Research
& Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh PA 15260

Dr Arthur M Glenberg
University of Wisconsin
W J Brodgen Psychology Bldg
1202 W Johnson Street
Madison WI 53708

Dr Marvin D Glock
13 Stone Hall
Cornell University
Ithaca NY 14853

Dr Sam Glucksberg
Department of Psychology
Princeton University
Princeton NJ 08540

Dr Joseph Goguen
Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park CA 94025

Dr Susan Goldman
University of California
Santa Barbara CA 93106

Dr Daniel Gopher
Industrial Engineering
& Management
TECHNION
Haifa 32000
ISRAEL

Dr Sherrie Gott
AFHRL/MDJ
Brooks AFB TX 78235

Jordan Grafman Ph D
2021 Lyttonsville Road
Silver Spring MD 20910

Dr Wayne Gray
Army Research Institute
5001 Eisenhower Avenue
Alexandria VA 22333

Dr Bert Green
Johns Hopkins University
Department of Psychology
Charles & 34th Street
Baltimore MD 21218

Dr James G Greene
University of California
Berkeley CA 94720

Prof Edward Haerfel
School of Education
Stanford University
Stanford CA 94305

Dr Henry M Hallff
Hallff Resources Inc
4918 33rd Road North
Arlington VA 22207

Janice Hart
Office of the Chief
of Naval Operations
OP-11HD
Department of the Navy
Washington D C 20350-2000

Mr William Hartung
PEAM Product Manager
Army Research Institute
5001 Eisenhower Avenue
Alexandria VA 22333

Dr Wayne Harvey
Center for Learning Technology
Educational Development Center
55 Chapel Street
Newton MA 02460

Prof John R Hayes
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh PA 15213

Dr Barbara Hayes-Roth
Department of Computer Science
Stanford University
Stanford CA 95305

Dr Joan I Heller
505 Haddon Road
Oakland CA 94608

Dr Shelly Heller
Department of Electrical Engi-
neering & Computer Science
George Washington University
Washington DC 20052

Dr Jim Hollan
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla CA 92093

Dr Melisse Holland
Army Research Institute for the
Behavioral and Social Science
5001 Eisenhower Avenue
Alexandria VA 22333

Ms Julie S Hough
Lawrence Erlbaum Associates
6012 Greene Street
Philadelphia PA 19144

Dr James Howard
Dept of Psychology
Human Performance Laboratory
Catholic University of
America
Washington DC 20064

Dr Earl Hunt Department of Psychology University of Washington Seattle WA 98105	Margaret Jerome c/o Dr Peter Chandler 63 The Drive Hove Sussex UNITED KINGDOM	Dr Paula Kirk Oakridge Associated Universities University Programs Division P O Box 117 Oakridge, TN 37831-0117	Dr Alan M Lesgold Learning Research and Development Center University of Pittsburgh Pittsburgh, PA 15260
Dr Ed Hutchins Intelligent Systems Group Institute for Cognitive Science (C-015) UCSD La Jolla, CA 92093	Dr Douglas H Jones Thatcher Jones Associates P O Box 6640 10 Trefalgar Court Lawrenceville, NJ 08648	Dr David Klehr Carnegie-Mellon University Department of Psychology Schenley Park Pittsburgh, PA 15213	Dr Jim Levin Department of Educational Psychology 210 Education Building 1310 South Sixth Street Champaign IL 61820-6990
Dr Dillon Inouye WICAT Education Institute Provo, UT 84057	Dr Marcel Just Carnegie Mellon University Department of Psychology Schenley Park Pittsburgh PA 15213	Dr Stephen Kosslyn Harvard University 1236 William James Hall 33 Kirkland St Cambridge MA 02138	Dr John Levine Learning R&D Center University of Pittsburgh Pittsburgh PA 15260
Dr Alice Isen Department of Psychology University of Maryland Collegeville MD 21228	Dr Ruth Kenfer University of Minnesota Department of Psychology Elliott Hall 75 E River Road Minneapolis MN 55455	Dr Kenneth Kotovsky Department of Psychology Community College of Allegheny County 800 Allegheny Avenue Pittsburgh, PA 15233	Dr Clayton Lewis University of Colorado Department of Computer Science Campus Box 430 Boulder CO 80309
Dr R J K Jacob Computer Science and Systems Code 7590 Information Technology Division Naval Research Laboratory Washington DC 20375	Dr Milton S Katz Army Research Institute 5001 Eisenhower Avenue Alexandria VA 22333	Dr Benjamin Kuipers University of Texas at Austin Department of Computer Sciences T S Painter Hall J 28 Austin TX 78712	Library Naval War College Newport RI 02940
Dr Zachary Jacobson Bureau of Management Consulting 365 Laurier Avenue West Ottawa Ontario K1A 0S5 CANADA	Dr Dennis Kibler University of California Department of Information and Computer Science Irvine CA 92717	Dr Pat Langley University of California Department of Information and Computer Science Irvine CA 92717	Library Naval Training Systems Center Orlando FL 32813
Dr Robert Jennerone Department of Psychology University of South Carolina Columbia, SC 29208	Dr David Kieras University of Michigan Technical Communication College of Engineering 1223 E Engineering Building Ann Arbor MI 48109	M Diane Langston Communications Design Center Carnegie-Mellon University Schenley Park Pittsburgh PA 15213	Dr Charlotte Linde Structural Semantics P O Box 707 Palo Alto CA 94320
Dr Claude Janvier Directeur, CIRADE Universite du Quebec a Montreal P O Box 8888 St A Montreal Quebec H3C 3P8 CANADA	Dr Peter Kincaid Training Analysis & Evaluation Group Department of the Navy Orlando FL 32813	Dr Jill Larkin Carnegie-Mellon University Department of Psychology Pittsburgh, PA 15213	Dr Marcia C Linn Lawrence Hall of Science University of California Berkeley CA 94720
Dr Robin Jeffries Hewlett-Packard Laboratories P O Box 10490 Palo Alto CA 94303-0971		Dr R W Lawler ARI 6 S 10 5001 Eisenhower Avenue Alexandria VA 22333 5600	Dr Frederic M Lord Educational Testing Service Princeton NJ 08541
		Dr Sandra P Marshall Dept of Psychology San Diego State University San Diego CA 92182	

Dr Richard E Mayer
Department of Psychology
University of California
Santa Barbara CA 93106

Dr Jay McClelland
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr Joe McLechlan
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr James S McMichael
Navy Personnel Research
and Development Center
Code 05
San Diego, CA 92152

Dr Barbara Means
Human Resources
Research Organization
1100 South Washington
Alexandria, VA 22314

Dr Arthur Melmed
U S Department of Education
724 Brown
Washington, DC 20208

Dr George A Miller
Department of Psychology
Green Hall
Princeton University
Princeton, NJ 08540

Dr James R Miller
MCC
9430 Research Blvd
Echelon Building #1, Suite 231
Austin TX 78759

Dr Mark Miller
Computer-Thought Corporation
1721 West Plano Parkway
Plano, TX 75075

Dr Andrew R Molnar
Scientific and Engineering
Personnel and Education
National Science Foundation
Washington DC 20550

Dr William Montague
MPRDC Code 13
San Diego, CA 92152-6800

Dr Randy Munaw
Program Manager
Training Research Division
HUMRRD
1100 S Washington
Alexandria, VA 22314

Dr Allen Munro
Behavioral Technology
Laboratories - USC
1845 S Elena Ave 4th Floor
Redondo Beach, CA 90277

Dr T Niblett
The Turing Institute
36 North Manover Street
Glasgow G1 2AD, Scotland
UNITED KINGDOM

Dr Richard E Nisbett
University of Michigan
Institute for Social Research
Room 5281
Ann Arbor MI 48109

Dr Mary Jo Nissen
University of Minnesota
N218 Elliott Hall
Minneapolis MN 55455

Dr A F Norcio
Computer Science and Systems
Code 7590
Information Technology Division
Naval Research Laboratory
Washington DC 20375

Dr Donald A Norman
Institute for Cognitive
Science C-015
University of California, San Diego
La Jolla California 92093

Director Training Laboratory
MPRDC (Code 05)
San Diego CA 92152-6800

Director, Manpower and Personnel
Laboratory
MPRDC (Code 06)
San Diego, CA 92152-6800

Director, Human Factors
& Organizational Systems Lab.
MPRDC (Code 07)
San Diego, CA 92152-6800

Library, MPRDC
Code P201L
San Diego, CA 92152-6800

Technical Director
Navy Personnel R&D Center
San Diego CA 92152-6800

Dr Harold F O'Neil Jr
School of Education - MPH 801
Department of Educational
Psychology & Technology
University of Southern California
Los Angeles, CA 90089-0031

Dr Michael Oberlin
Naval Training Systems Center
Code 711
Orlando FL 32813-7100

Dr Stellan Ohlsson
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh PA 15213

Director, Research Programs
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Office of Naval Research
Code 1133
800 N Quincy Street
Arlington, VA 22217-5000

Office of Naval Research
Code 1142PS
800 N Quincy Street
Arlington, VA 22217-5000

Office of Naval Research
Code 1142CS
800 N Quincy Street
Arlington, VA 22217-5000
(6 Copies)

Office of Naval Research
Code 11R
800 N Quincy Street
Arlington, VA 22217-5000

Director, Technology Programs
Office of Naval Research
Code 12
800 North Quincy Street
Arlington, VA 22217-5000

Office of Naval Research
Code 125
800 N Quincy Street
Arlington, VA 22217-5000

Psychologist
Office of Naval Research
Branch Office, London
Box 39
FPO New York NY 09510

Special Assistant for Marine
Corps Matters
ONR Code 00MC
800 N Quincy St
Arlington VA 22217-5000

Psychologist
Office of Naval Research
Liaison Office, Far East
APO San Francisco CA 96503
Office of Naval Research
Resident Representative
UCSD

University of California,
San Diego
La Jolla CA 92093-0001

Assistant for Planning MANTR,
OP 01B6
Washington DC 20370

Assistant for MPT Research
Development and Studies
OP 0187
Washington, DC 20370

Dr Judith Orasenu
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

CDR R T Parlette
Chief of Naval Operations
OP-112G
Washington DC 20370-2000

Dr James Paulson
Department of Psychology
Portland State University
P O Box 751
Portland, OR 97207

Dr Douglas Pearce
DCIEM
Box 2000
Downsview, Ontario
CANADA

Dr James W Pellegrino
University of California
Santa Barbara
Department of Psychology
Santa Barbara CA 93106

Dr Virginia E Pendergrass
Code 711
Naval Training Systems Center
Orlando, FL 32813-7100

Dr Nancy Pennington
University of Chicago
Graduate School of Business
1101 E 58th St
Chicago, IL 60637

Military Assistant for Training and
Personnel Technology
OUSD (R & E)
Room 3D129 The Pentagon
Washington DC 20301 3080

Dr Ray Perez
ARI (PERI-11)
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr David N Perkins
Educational Technology Center
337 Gulman Library
Appian Way
Cambridge, MA 02138

Dr Steven Pinker
Department of Psychology
EtO-018
MIT
Cambridge, MA 02139

Dr Tjeerd Plomp
Twente University of Technology
Department of Education
P O Box 217
7500 AE ENSCHEDE
THE NETHERLANDS

Dr Martha Polson
Department of Psychology
Campus Box 346
University of Colorado
Boulder CO 80309

Dr Peter Polson
University of Colorado
Department of Psychology
Boulder CO 80309

Dr Michael I Posner
Department of Neurology
Washington University
Medical School
St Louis MO 63110

Dr Joseph Psotka
ATTN PERI-1C
Army Research Institute
5001 Eisenhower Ave
Alexandria, VA 22333

Dr Mark D Reckase
ACT
P O Box 168
Iowa City IA 52243

Dr Lynne Reder
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburg, PA 15213

Dr Wesley Regian
AFHRL/MOD
Brooks AFB, TX 78235

Dr Fred Reif
Physics Department
University of California
Berkeley, CA 94720

Dr Lauren Resnick
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Dr Gil Ricard
Mail Stop C04-14
Grumman Aerospace Corp
Bethpage, NY 11714

Mark Richer
1041 Lake Street
San Francisco, CA 94118

Dr Linda G Robertis
Science, Education, and
Transportation Program
Office of Technology Assessment
Congress of the United States
Washington DC 20510

Dr Andrew M Rose
American Institutes
for Research
1055 Thomas Jefferson St NW
Washington DC 20007

Dr David Rumlhart
Center for Human
Information Processing
Univ of California
La Jolla, CA 92093

Dr James F Sanford
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr Walter Schneider
Learning R&D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr Alan H Schoenfeld
University of California
Department of Education
Berkeley CA 94720

Dr Janet Schofield
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Karen A Schriver
Department of English
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr Marc Sebrechts
Department of Psychology
Wesleyan University
Middletown CT 06475

Dr Judith Segal
OERI
555 New Jersey Ave NW
Washington, DC 20208

Dr Colleen M Seifert
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla CA 92093

Dr Ramsey W Selden
Assessment Center
CCSSO
Suite 379
400 N Capitol NW
Washington DC 20001

Dr. Sylvia A. S. Shaflo
Department of
Computer Science
Towson State University
Towson, MD 21204

Dr. Ben Shneiderman
Dept. of Computer Science
University of Maryland
College Park, MD 20742

Dr. Lee Shulman
Stanford University
1040 Cathcart Way
Stanford, CA 94305

Dr. Randall Shumaker
Naval Research Laboratory
Code 7510
4555 Overlook Avenue S.W.
Washington, DC 20375-5000

Dr. Valerie Shute
AFHRL/MOE
Brooks AFB, TX 78235

Dr. Robert S. Siegler
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Zita M. Simutis
Instructional Technology
Systems Area
ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. H. Wallace Sinsko
Manpower Research
and Advisory Services
Smithsonian Institution
801 North Pitt Street
Alexandria, VA 22314

Dr. Derek Sleeman
Dept. of Computing Science
King's College
Old Aberdeen
AB9 2UB
UNITED KINGDOM

Dr. Richard E. Snow
Department of Psychology
Stanford University
Stanford, CA 94306

Dr. Elliot Soloway
Yale University
Computer Science Department
P.O. Box 2158
New Haven, CT 06520

Dr. Kathryn T. Spoehr
Brown University
Department of Psychology
Providence, RI 02912

James J. Staszewski
Research Associate
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Robert Sternberg
Department of Psychology
Yale University
Box 11A, Yale Station
New Haven, CT 06520

Dr. Albert Stevens
Bolt Beranek & Newman, Inc.
10 Moulton St.
Cambridge, MA 02238

Dr. Paul J. Stiche
Senior Staff Scientist
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Thomas Sticht
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. John Tangney
AFOSR/NL
Bolling AFB, DC 20332

Dr. Kikumi Tatsuoka
CERL
252 Engineering Research
Laboratory
Urbana, IL 61801

Dr. Robert P. Taylor
Teachers College
Columbia University
New York, NY 10027

Dr. Perry W. Thorndyke
PMC Corporation
Central Engineering Labs
1185 Coleman Avenue, Box 580
Santa Clara, CA 95052

Dr. Sharon Tkacz
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Douglas Towne
Behavioral Technology Labs
1845 S. Elena Ave.
Redondo Beach, CA 90277

Dr. Paul Twohig
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Kurt Van Lehn
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Jerry Vogt
Navy Personnel R&D Center
Code 51
San Diego, CA 92152-6800

Dr. Beth Warren
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. Barbara White
Bolt Beranek & Newman, Inc.
10 Moulton Street
Cambridge, MA 02238

LCDR Cory deGroot Whitehead
Chief of Naval Operations
OP-112G1
Washington, DC 20370-2000

Dr. Heather Wild
Naval Air Development
Center
Code 6021
Warminster, PA 18974-5000

Dr. William Clancey
Stanford University
Knowledge Systems Laboratory
701 Welch Road, Bldg. C
Palo Alto, CA 94304

Dr. Michael Williams
IntelliCorp
1975 El Camino Real West
Mountain View, CA 94040-2218

Dr. Robert A. Wisher
U.S. Army Institute for the
Behavioral and Social Science
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Martin F. Wiskoff
Navy Personnel R & D Center
San Diego, CA 92152-6800

Dr. Dan Wolz
AFHRL/MOE
Brooks AFB, TX 78235

Dr. Wallace Wulfbeck, III
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Joe Yasutake
AFHRL/LRT
Lowry AFB, CO 80230

Dr. Joseph L. Young
Memory & Cognitive
Processes
National Science Foundation
Washington, DC 20550

Dr. Steven Zornetzer
Office of Naval Research
Code 114
800 N. Quincy St.
Arlington, VA 22217-3000

END

DATE

FILMED

JAN

1988